CSL 302 Lab 5 Report

Piyush Jain
2015CSB1023

1) Value Iterations

Value iteration is a method of computing an optimal MDP policy and its value.

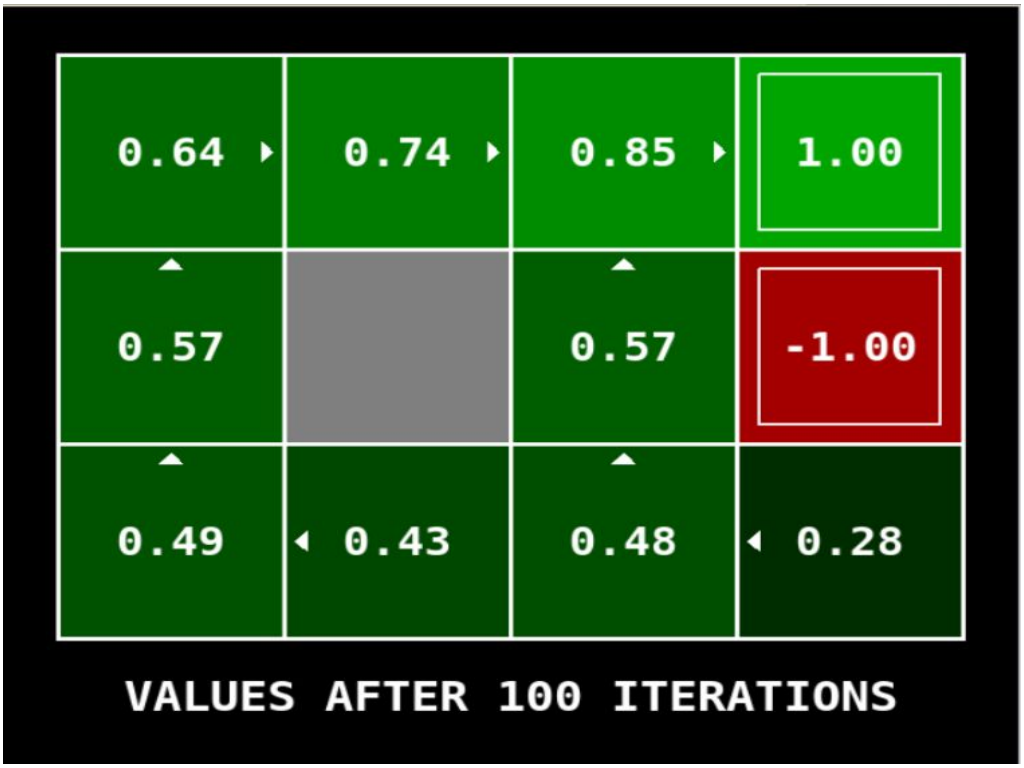
Value iteration starts at the "end" and then works backward, refining an estimate of either Q* or V*. There is really no end, so it uses an arbitrary end point. Let Vk be the value function assuming there are k stages to go, and let Qk be the Q-function assuming there are k stages to go. These can be defined recursively. Value iteration starts with an arbitrary function V0 and uses the following equations to get the functions for k+1 stages to go from the functions for k stages to go:

$$Q_{k+1}(s,a) = \sum_{s'} P(s'|s,a)\,(R(s,a,s') + \gamma V_k(s')) \quad \text{for } k \geq 0$$
$$V_k(s) = \max_a Q_k(s,a) \quad \text{for } k>0.$$

Observations

Below are the figures of value iterations in gridworld after 100 iterations, first figure is q -value and second figure is v value for state. Note that V values will always be maximum value for given state.


0.59
0.57
0.64
0.53
0.67
0.60
0.74
0.67
0.77
0.66
0.85
0.57
1.00
0.57
0.51
0.51
0.46
0.57
0.53
-0.60
0.30
-1.00
0.49
0.45
0.41
0.44
0.40
0.43
0.42
0.40
0.48
0.40
0.29
0.41
-0.65
0.28
0.13
0.27
Q-VALUES AFTER 100 ITERATIONS


0.64
0.74
0.85
1.00
0.57
0.57
-1.00
0.49
0.43
0.48
0.28
VALUES AFTER 100 ITERATIONS

2) Bridge Crossing Analysis

Bridge Grid is a grid world map with the a low-reward terminal state and a high-reward terminal state separated by a narrow "bridge", on either side of which is a chasm of high negative reward. We have to change either discount or noise so that in optimal policy agent can cross the bridge.

Observations

answerDiscount = 0.9
answerNoise = 0

3) Policies

This question is consist of discount grid consisting of two paths 1) paths that "risk the cliff" and travel near the bottom row of the grid; these paths are shorter but risk earning a large negative payoff, and are represented by the red arrow in the figure below. (2) paths that "avoid the cliff" and travel along the top edge of the grid. These paths

are longer but are less likely to incur huge negative payoffs.We have to choose discount value , living reward and noise

Observations

a) Prefer the close exit(+1), risking the cliff (-10)
   Discount = 0.9
   Noise = 0.0
   Living Reward = -5.0

   Note that if we want to risk the cliff then we should have high negative living reward so that agent takes shortest path (risking the cliff).

b) Prefer the close exit(+1), avoiding the cliff (-10)
   Discount = 0.6
   Noise = 0.4
   Living Reward = -1.0

   Note that now to avoid cliff we have to reduce the living reward and decreases the discount as well so that it takes (+1) exit rather (+10) exit.

c) Prefer the distant exit (+10), risking the cliff (-10)
   Discount = 0.9
   Noise = 0.0
   Living Reward = -1.0

   Note that if we want to take the distant exit then we have to increase the discount value so that it takes distant exit.But here it will risk the cliff as due to negative living reward.

d) Prefer the distant exit(+1), avoiding the cliff (-10)
   Discount = 0.9

Noise = 0.0
Living Reward = 0.0

Note that to avoid the cliff then we should have no negative living reward so that agent can takes distant path (avoiding the cliff).

e) Avoid exit(+1) and exit (+10) and cliff (-10)
Discount = 1.0
Noise = 0.0
Living Reward = 15.0

Note that to avoid game exit it should have high living reward so that game continues forever.

4) **Q- Learning**

Q-learning is a reinforcement learning technique used in machine learning. The technique does not require a model of the environment. Q-learning can handle problems with stochastic transitions and rewards, without requiring adaptations.

In this question we perform action and according to that we update q values using q update equation.

5) Epsilon - greedy

In this we have implemented functions which is exploration probability. In getAction method we flip the coin if the probability outcome is less than exploration probability then we choose random action so that agent can explore and if it is more than the flip coin outcome then we choose best action depending upon their q values.

Observations

In previous question there is no scope for exploration so agent can not learn efficiently but in this question we are taking random action so that it can explore more.

6) Q-Learning and Pacman

This is basically testing of above the question. In this we play pacman game, Pacman will play games in two phases. In the first phase, training, Pacman will begin to learn about the values of positions and actions. Because it takes a very long time to learn accurate Q-values even for tiny grids. Once Pacman's training is complete, he will enter testing mode. When testing, Pacman's self.epsilon and self.alpha will be set to 0.0, effectively stopping Q-learning and disabling exploration, in order to allow Pacman to exploit his learned policy.

Here we take 2000 games for training and after that testing.

Observations

Here are the score of first 10 training games of pacman in which pacman

Average Score: -506.9

Scores: -507.0, -506.0, -508.0, -504.0, -507.0, -508.0, -511.0, -506.0, -506.0, -506.0

Win Rate: 0/10 (0.00)

Record: Loss, Loss, Loss, Loss, Loss, Loss, Loss, Loss, Loss, Loss